UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/624,831	07/25/2000	Steven E. Baker	58689-8060.US01	7340

57284 7590 09/08/2006

PERKINS COLE LLP
101 JEFFERSON DRIVE
MENLO PARK, CA 94025

EXAMINER	
FOWLKES, ANDRE R	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 09/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/624,831	BAKER, STEVEN E.	
	Examiner	Art Unit	
	Andre R. Fowlkes	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 June 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-7 and 13-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7 and 13-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 6/12/06.

2. Claims 1-7 and 13-22 are pending. Claims 1 and 5 have been amended. New claims 13-22 have been added.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claim 5 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Specifically, there is no support given, from the original disclosure, for claim 5, amended on 6/10/05. There appears to be a typo, in that "recursively simulating and" should be removed from claim 5, section c. At p. 15, lines 1-6 of the specification, support is provided for simulating a main batch file that begins a recursive process... in that scripts, sub-batch files or procedures called by the initial batch file, in turn, can call other programs, procedures, scripts and sub-batch files. However, claim 5, section c

describes the recursive simulation... of the execution of the second process and claim 5, section b discloses that the second process, alone, results in a simulation being performed. Sections b and c of claim 5, in combination, perform the simulation of a file that in itself performs a simulation. This situation does not appear to be described or implied in the specification or arguments. In the interests of compact prosecution, the examiner is interpreting "recursively simulating" as the process of simulating a file that contains recursive procedures, as described in applicant's specification.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-7 and 13-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Amberg et al., (Amberg), U.S Patent No. 5,991,543 in view of Rickel et al., (Rickel), U.S. Patent No. 5,854,924, further in view of Papachristou et al., (Papachristou), "Microprocessor Based Testing for Core-Based System on a Chip", 1999, ACM 1-58113-092-9/99/0006 (art made of record).

As per claim 1, Amberg discloses:

- a method of testing a process that downloads and installs customer ordered software onto a target computer (abstract lines 1-2, "A method for installing and/or testing software for build to order computer systems").

- dynamically generating a file that includes instructions that when executed launch the process of downloading and installing customer ordered software, including a combination of files from a library, to a target computer (col. 3 line 66 – col. 4 line 17, "Step maker 140 is a computer system configured to sequence the software installation ... steps to be run on target system 160. To sequence the software installation ... steps, step maker 140, and more particularly, sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96... sequencing program 204 retrieves a plurality of software installation ... steps corresponding to the component descriptors ... over (the) network connection 110... Having retrieved the software installation ... steps appropriate for target system 160, sequencing program 204 sequences the steps ... the output files include text files containing command lines appropriate for executing the appropriate software installation ... steps upon target system", and software packages include a combination of library files).

Amberg does not explicitly disclose:

- on a simulation computer, simulating the execution of said dynamically generated file in accordance with a set of evaluation rules such that the outcome of the execution of said file is determined,

- simulating an environment in which the combination of files from the library run in and interact with,

- analyzing the outcome of the simulation of the execution of said dynamically generated file to determine possible syntax errors and possible flow errors.

However, Rickel, in an analogous environment, discloses:

- **on a simulation computer** (col. 7 lines 46-47, "code for the intermediate file 16 is generated", and figure 1a shows the intermediate file within the debugging system. Figure 1b and the associated text (e.g. col. 2 lines 45-47) shows the computer that is used to store and use the debugging tool), **simulating the execution of said dynamically generated file in accordance with a set of evaluation rules such that the outcome of the execution of said file is determined.** (col. 1 line 55 – col. 2 line 9, "The static debugging tool includes an analyzer for causing the computer to statically analyze (i.e. simulate) a representation of a ... file to detect the presence of program errors... without executing the ... file ... the debugging tool may include a system call and restrictions library (i.e. rules file) for providing information to the static debugging tool which is specific to particular system that the ... file is designed to be used"),

- simulating an environment in which the combination of files from the library run in and interact with (col. 2 lines 19-29, "The analyzer detects the errors and potential errors in the ... file by following all of the possible flow paths (i.e. to and from library components) of the ... file while tracking the use of various program parameters ... These various program parameters ... may include checking for ...

inconsistent use of a certain variable type", and col. 2:5-12, "In another embodiment, the debugging tool may include a system call and restrictions library file for providing information to the static debugging tool which is specific to a particular system (i.e. Environment) that the binary program file is designed to be used. This system call and restrictions library may include certain constraints and parameter definitions that are specific to a particular system on which the binary program file is intended to be used", and col. 4:5-7, "This information may include ... details about global data functions"),

- and **analyzing the outcome of the simulation of the execution of said dynamically generated file to determine possible syntax errors and possible flow errors** (col. 2 lines 19-29, "The analyzer detects the errors and potential errors in the ... file by following all of the possible flow paths of the ... file while tracking the use of various program parameters ... These various program parameters ... may include checking for ... inconsistent use of a certain variable type (i.e. syntax errors)").

Therefore, it would have been obvious to a person of ordinary skill in that art at the time the invention was made to incorporate the teachings of Rickel into the teachings of Amberg to:

- on a simulation computer, simulating the execution of said dynamically generated file in accordance with a set of evaluation rules such that the outcome of the execution of said file is determined,

- simulating an environment in which the combination of files from the library run in and interact with,

- analyzing the outcome of the simulation of the execution of said dynamically generated file to determine possible syntax errors and possible flow errors.

The modification would have been obvious because one of ordinary skill in the art would want to save time and testing costs by generating and testing the file on the simulation computer. One of ordinary skill in the art would have also been motivated to use rules to simulate the execution of the dynamically generated file in order to safely test many aspects of the file, without having to actually execute the file. Additionally, one of ordinary skill in the art would have wanted to analyze the outcome of the simulation of the execution of said file in order to find and correct possible syntax and flow errors in order to produce a defect-free file without the risk or expense of actually executing the file to identify the errors.

The Amberg/Rickel combination doesn't explicitly disclose **simulating the process of downloading** a file. However, Papachristou, in an analogous environment, discloses **simulating the process of downloading** a file, (p. 590 col. L:13, "simulating the download").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Papachristou into the Amberg/Rickel combination to **simulating the process of downloading** a file. The modification would have been obvious because one of ordinary skill in the art would have wanted to estimate the accuracy of and optimize the speed of a download.

As per claim 2, the rejection of claim 1 is incorporated and further Amberg discloses that **said dynamically generated file is a main batch file created from a static text file that indicates the model types of the computer a lookup file that indicates the necessary instruction required to be executed for the model type indicated, and a process that reads the model type from said static text file and creates said dynamically generated file by reading said lookup file to determine command components** (col. 3 line 51 – col. 4 line 17, "To sequence the software installation ... steps, ... (a) sequencing program ... reads a plurality of component descriptors ... Component descriptors are computer readable descriptions of the components of (the) target system ... Having sequenced the steps required for target system 160, sequencing program 204 writes a series of ... files ... the output files include text files containing command lines (batch files) appropriate for executing the appropriate software installation ... steps upon target system").

As per claim 3, the rejection of claim 2 is incorporated and further Amberg discloses that the **main batch file contains one or more labels identifying the flow of the process** (abstract line 10, "creating a file including a start of execution indication (flow label)"), **and one or more commands containing instructions to be executed and one or more calls to one or more static batch files** (col. 12 lines 57-58, "Batch file (an ASCII text file containing a sequence of commands) 870 is then run").

As per claim 4, the rejection of claim 3 is incorporated and further Amberg discloses that the **process of simulating said dynamically generated batch file comprises recursively simulating each of said one or more batch files to determine the outcome of the process** (Ambergs batch file is a recursive program. Figure 10 shows the routine Runstep.exe (note that .bat and .exe files are both executable files) being called as a subroutine by the Runstep.bat routine that Runstep.exe called itself. Rickel is used to simulate the batch file created by Amberg, as addressed above in claim 1, from which claim 4 ultimately depends. The Amberg/Rickel/Houghton combination; therefore, provides the recursive simulation of the dynamically generated batch file.)

As per claim 5, this is a system version of the claimed method discussed above, in claim 4, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Amberg Fig. 10, Rickel col. 1 line 55 – col. 2 line 9 and Papachristou, p. 590 col. L:13.

As per claim 6, the rejection of claim 5 is incorporated and further Amberg discloses that the **first process reads a electronic traveler to determine the model of the target computer , looks up in the master token list the model of the target computer and creates from the information in the master token list a second process that is an executable main batch file that downloads and installs customer ordered computer software onto the target computer** (col. 3 line 51 –

col. 4 line 17, "To sequence the software installation ... steps, ... (a) sequencing program (first process) ... reads a plurality of component descriptors (electronic traveler) ... Component descriptors are computer readable descriptions of the components of (the) target system ... Having read the ... component descriptors, sequencing program **204** retrieves ... software installation ... steps corresponding to the component descriptors from the database (master token list) ... Having sequenced the steps required for target system 160, sequencing program 204 writes a series of ... files ... the output files include text files containing command lines (executable batch files) appropriate for executing the appropriate software installation ... steps upon target (computer) system").

As per claim 7, the rejection of claim 6 is incorporated and further Amberg discloses that **said batch file contains labels, commands, and sub batch file calls** (abstract line 10, "creating a file including a start of execution indication (flow label)", and col. 12 lines 57-58, "Batch file (an ASCII text file containing a sequence of commands) 870 is then run").

Amberg doesn't explicitly disclose that **said third process interpretively tracks said labels, simulates each of said commands and recursively evaluates each of said sub batch files until the end of the main batch file is reached by said third process.**

However, Rickel, in an analogous environment, discloses:

- that a **process interpretively tracks said labels** (abstract lines 9 – 11, "the analyzer detects the errors... by following all of the possible flow paths", and the flow paths are labeled, col. 4 lines 62-63, "file 16 includes information (labels) identifying the function flow paths").

- **simulates each of said commands** (col. 2 lines 30 – 32, "the static debugging tool symbolically executes ... (the) file").

- and recursively evaluated each of said sub batch files until the end of the main batch file is reached (col. 2 lines 16 - 21, "the (sub batch file) calls within the ... file (are represented symbolically) ... the analyzer detects the errors ... (in the) file by following all of the possible flow paths (recursive as well as iterative) of the ... file").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Rickel into the teachings of Amberg to have a system wherein the main batch file contains labels, commands, and sub batch file calls, and a third process that interpretively tracks the labels, simulates each of the commands and recursively evaluates each of the sub batch files. The modification would have been obvious because one of ordinary skill in the art would be motivated to have robust method of detecting errors in software capable of using the labels in the software to produce detailed error reports.

As per claim 13, the rejection of claim 1 is incorporated and further Amberg discloses **reporting said syntax errors and flow errors in a readable format** (col. 14 lines 22-26, "results from the installation and testing may be logged ... the results

preferably include whether all the steps were completed successfully and what types of failures (i.e. syntax errors and flow errors) ... were encountered").

As per claim 14, the rejection of claim 1 is incorporated and further Amberg discloses **simulating an entire test process** (col. 14 lines 22-26, "results from the installation and testing may be logged ... the results preferably include whether all the steps were completed successfully and what types of failures ... were encountered").

As per claim 15, the rejection of claim 1 is incorporated and further Amberg discloses **analyzing an entire simulated test process** (col. 14 lines 22-26, "results from the installation and testing may be logged ... the results preferably include whether all the steps were completed successfully and what types of failures ... were encountered").

As per claim 16, the rejection of claim 1 is incorporated and further Amberg discloses **simulating a download process** (col. 14 lines 22-26, "results from the installation and testing may be logged ... the results preferably include whether all the steps were completed successfully and what types of failures ... were encountered").

As per claim 17, the rejection of claim 1 is incorporated and further Amberg discloses **simulating a chip programming process** (col. 1:37-65, "In general, testing detects and analyzes errors that occur in both the hardware and software portions of the

computer system. A partial list of computer system hardware tests might include diagnostics upon hardware components such as a processor, memory, a disk storage device, an audio device, a graphics device, a keyboard, a mouse, and a printer. Software installation often includes loading a desired package of software onto the computer system (i.e. chip), preparing appropriate environment variables for the computer, and preparing appropriate initialization files for the loaded software. Software testing often includes making sure that a desired version of software has been installed onto the computer system and that appropriate drivers are present on the computer system").

As per claim 18, the rejection of claim 1 is incorporated and further Amberg discloses **determining what an outcome of a hypothetical execution would be in a specific simulated environment** (col. 1:37-65, "In general, testing detects and analyzes errors that occur in both the hardware and software portions of the computer system. A partial list of computer system hardware tests might include diagnostics upon hardware components such as a processor, memory, a disk storage device, an audio device, a graphics device, a keyboard, a mouse, and a printer. Software installation often includes loading a desired package of software onto the computer system, preparing appropriate environment variables for the computer, and preparing appropriate initialization files for the loaded software. Software testing often includes making sure that a desired version of software has been installed onto the computer system and that appropriate drivers are present on the computer system.

It has been known in the industry to install software and to test computer systems during manufacture by performing a fixed procedure before they are shipped to customers. For instance, a diskette containing certain diagnostic tests for a certain type of computer system is created. The diskette includes lengthy, often-complicated batch files which direct the software installation and diagnostic processes. The diskette further contains all the executable files for performing tests on the computer system being purchased").

As per claim 19, the rejection of claim 1 is incorporated and further Amberg discloses **taking less than one hour to perform said dynamically generating, said simulating execution, said simulating an environment, and said analyzing the outcome of the simulation** (col. 7:5-10, "More specifically, the step file allows commands to be repeated for a defined number or iterations or for a defined length of time.").

As per claim 20, the rejection of claim 1 is incorporated and further Amberg discloses **cleaning up errors to facilitate code re-use** (col. 1:37-65, "In general, testing detects and analyzes errors that occur in both the hardware and software portions of the computer system. A partial list of computer system hardware tests might include diagnostics upon hardware components such as a processor, memory, a disk storage device, an audio device, a graphics device, a keyboard, a mouse, and a printer. Software installation often includes loading a desired package of software onto the

computer system, preparing appropriate environment variables for the computer, and preparing appropriate initialization files for the loaded software. Software testing often includes making sure that a desired version of software has been installed onto the computer system and that appropriate drivers are present on the computer system.

It has been known in the industry to install software and to test computer systems during manufacture by performing a fixed procedure before they are shipped to customers. For instance, a diskette containing certain diagnostic tests for a certain type of computer system is created. The diskette includes lengthy, often-complicated batch files which direct the software installation and diagnostic processes. The diskette further contains all the executable files for performing tests on the computer system being purchased").

As per claim 21, the rejection of claim 5 is incorporated and further Amberg discloses **an expert tool for facilitating management of software libraries responsible for testing many computers per day** (col. 1:55-65, "It has been known in the industry to install software and to test computer systems during manufacture by performing a fixed procedure before they are shipped to customers. For instance, a diskette containing certain diagnostic tests for a certain type of computer system is created. The diskette includes lengthy, often-complicated batch files which direct the software installation and diagnostic processes. The diskette further contains all the executable files for performing tests on the computer system being purchased").

As per claim 22, the rejection of claim 5 is incorporated and further Amberg discloses that **the second process facilitates examination of batch files in a simulated environment** (col. 1:37-65, "In general, testing detects and analyzes errors that occur in both the hardware and software portions of the computer system. A partial list of computer system hardware tests might include diagnostics upon hardware components such as a processor, memory, a disk storage device, an audio device, a graphics device, a keyboard, a mouse, and a printer. Software installation often includes loading a desired package of software onto the computer system, preparing appropriate environment variables for the computer, and preparing appropriate initialization files for the loaded software. Software testing often includes making sure that a desired version of software has been installed onto the computer system and that appropriate drivers are present on the computer system.

It has been known in the industry to install software and to test computer systems during manufacture by performing a fixed procedure before they are shipped to customers. For instance, a diskette containing certain diagnostic tests for a certain type of computer system is created. The diskette includes lengthy, often-complicated batch files which direct the software installation and diagnostic processes. The diskette further contains all the executable files for performing tests on the computer system being purchased").

Response to Arguments

4. Applicants arguments have been considered but they are not persuasive.

In the remarks, the applicant has argued substantially that:

1) Amberg does not recursively examine batch files, at p. 5:24-25.

Examiner's response:

1) In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The Amberg art is used to disclose a recursive batch file. While the Rickel art is used to disclose the simulation of Ambergs recursive batch file.

In the remarks, the applicant has argued substantially that:

2) Rickel does not disclose examining multiple files for recursive examination from libraries to determine which ones would be executed in the factory, at p. 6:2-4.

Examiner's response:

2) In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., examining multiple files for recursive examination from libraries to determine which ones would be executed in the factory) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification

are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In the remarks, the applicant has argued substantially that:

3) Rickel does not disclose monitoring what multiple files would do with the resources of an execution environment, at p.6:4-6.

Examiner's response:

3) The examiner disagrees with applicants characterization of the applied art. Rickel does disclose simulation of the execution of a software application in a specific environment, col. 2:5-12, "In another embodiment, the debugging tool may include a system call and restrictions library file for providing information to the static debugging tool which is specific to a particular system (i.e. Environment) that the binary program file is designed to be used. This system call and restrictions library may include certain constraints and parameter definitions that are specific to a particular system on which the binary program file is intended to be used", and col. 4:5-7, "This information may include ... details about global data functions." Additionally, it is readily known in the art that a single software application is composed of multiple files.

In the remarks, the applicant has argued substantially that:

4) Houghton does not disclose simulating an environment or the recursive examination of batch files, at p. 6:7-8.

Examiner's response:

4) Applicant's arguments with respect to the Houghton art have been considered but are moot in view of the new ground(s) of rejection.

In the remarks, the applicant has argued substantially that:

5) Amberg does not disclose simulating the execution of a dynamically generated file, at p. 6:17-18 and p. 8:11-12.

Examiner's response:

5) In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The Amberg art is used to disclose creating a dynamically generated installation file. While the Rickel art is used to disclose the simulation of Ambergs installation file.

In the remarks, the applicant has argued substantially that:

6) Rickel does not disclose simulating the execution of a dynamically generated file that downloads and installs a combination of files, at p. 6:25-7:3 and p. 8:19-20.

Examiner's response:

6) In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The Amberg art is used to disclose creating a dynamically generated installation file. While the Rickel art is used to disclose the simulation of Ambergs installation file. The Papachristou art is used to disclose the simulation of a download process.

In the remarks, the applicant has argued substantially that:

7) Houghton does not disclose simulating the execution of a dynamically generated file, at p. 7:4-5.

Examiner's response:

7) Applicant's arguments with respect to the Houghton art have been considered but are moot in view of the new ground(s) of rejection.

In the remarks, the applicant has argued substantially that:

8) Houghton does not disclose simulating the process of downloading a file, at p. 7:12-13.

Examiner's response:

8) Applicant's arguments with respect to the Houghton art have been considered but are moot in view of the new ground(s) of rejection.

In the remarks, the applicant has argued substantially that:

9) Neither Amberg nor Rickel teach simulating an environment in which the combination of files from the library run in and interact with, at p. 7:22-27

Examiner's response:

9) The examiner disagrees with applicant's interpretation of the applied art. Rickel discloses simulating an environment in which the combination of files from the library run in and interact with, at col. 2:5-12, "In another embodiment, the debugging tool may include a system call and restrictions library file for providing information to the static debugging tool which is specific to a particular system (i.e. Environment) that the binary program file is designed to be used. This system call and restrictions library may include certain constraints and parameter definitions that are specific to a particular system on which the binary program file is intended to be used", and col. 4:5-7, "This information may include ... details about global data functions." Additionally, it is readily known in the art that a single software application is composed of multiple files.

In the remarks, the applicant has argued substantially that:

10) The applied art does not recursively simulate and interpret the outcome of the execution of a process that downloads and installs software, at p. 8:26-30.

Examiner's response:

10) The examiner disagrees with applicant's interpretation of the applied art. The Amberg/Rickel/Papachristou combination discloses recursive file simulation and interpretation of the outcome of the execution of a process that downloads and installs software, as addressed in the above art rejection. For example, see Amberg Fig. 10, Rickel col. 1 line 55 – col. 2 line 9 and Papachristou p. 590 col. L:13.

At p. 15, lines 1-6 of the specification, support is provided for simulating a main batch file that begins a recursive process... in that scripts, sub-batch files or procedures called by the initial batch file, in turn, can call other programs, procedures, scripts and sub-batch files. However, recursively performing a simulation does not appear to be described or implied in the specification or arguments. In the interests of compact prosecution, the examiner is interpreting "recursively simulating" as the process of simulating a file that contains recursive procedures, as described in applicant's specification.

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ARF

TUAN DAM
SUPERVISORY PATENT EXAMINER